Ra Cohen

ClusterTales

26th June 2023

BACKGROUND

Content Recommendations is a $2.1 billion industry as of 2020 and is only expected to grow exponentially as streaming services become ubiquitous. However, recommending content is a difficult data science problem especially without data on how users interact with the content which would enable approaches such as collaborative filtering. Developing a methodology to recommend content without this user data would enable new platforms without data to provide good recommendations and also could provide recommendation solutions to data privacy conscious areas such as the EU.

RESEARCH QUESTION

- Can we produce recommendations without any user-data based instead on a deeper understanding of the content itself?

DATASETS

My data consists primarily of data scraped from TVTropes and IMDB by myself and a prior research group (github). TVTropes is a wiki detailing tropes. A trope is any literary or rhetorical device that consists in the use of words in other than their literal sense or in other words a storytelling shorthand for a concept that the audience will recognize and understand instantly such as Smart People Play Chess. The existing dataset contained a list of all tropes, the description of each trope, each trope's usage in media, the narrative example of each usage, and the media's matching id in the IMDb dataset. I used these IMDb ids to scrape posters for my user interface. I also scraped TVTropes for how tropes related to each other via linkages in the descriptions to build a graph of the trope landscape. This data was aggregated in real-time into Neo4j as a graph and then exported into a list of nodes and edges to be reconstructed in python.

DATA CLEANING

Preprocessing

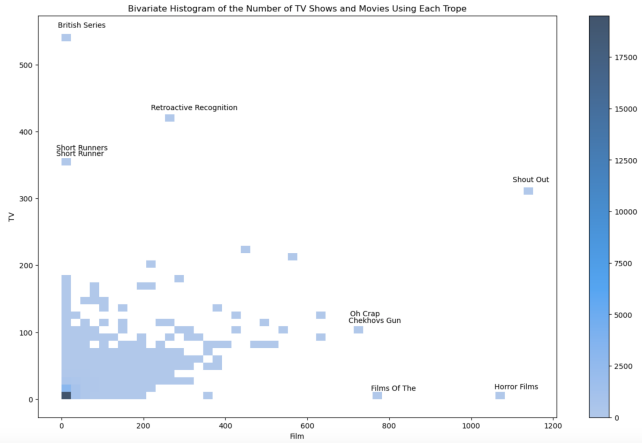
First I normalized the names of all of the tropes from their link format (SmartPeoplePlayChess) into the expected format (Smart People Play Chess). Then, I constructed my trope graph in python from the data I exported from my Neo4J instance. I dropped any tropes from my graph which were not present in the existing dataset.

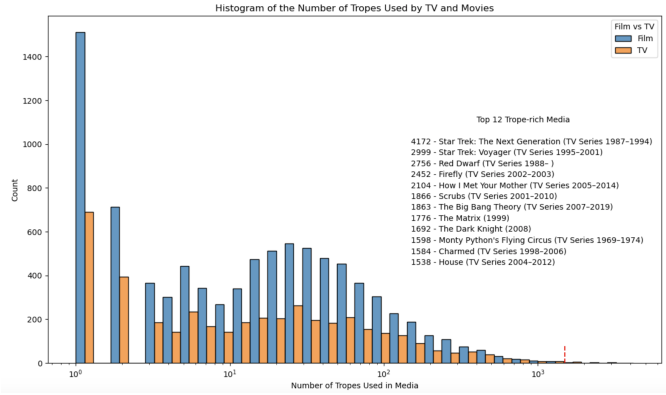
Data Transformation

Next, I converted the trope examples dataset into several dictionaries including mapping media to their list of tropes and tropes to their encapsulating media. I then used my scraped IMDb title information to create dictionaries for quickly converting between the english title for a piece of content and its associate id. Lastly, I constructed a dataset mapping every content to a vector of all tropes with a True if the trope is present in the content and a False if not.

VISUALIZATION

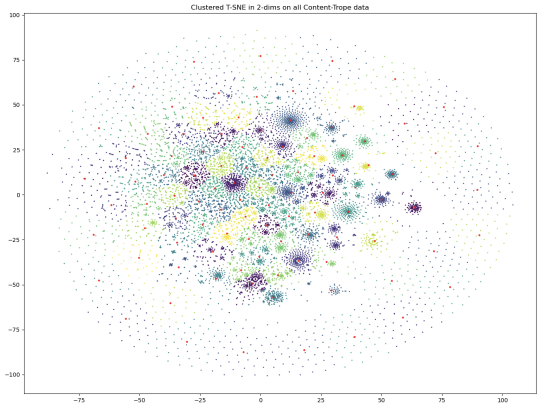
First I examined the number of media utilizing each trope to examine if there were specific tropes that were TV only or film only. I observed there were a large number of tropes (>18000) with only 1 documented example as shown to the right. Examining the outliers it became clear that there were a few oddities such as ‘Short Runner’ and ‘Short Runners’ being two distinct tropes and the ‘Films of the’ trope being listed as one conglomerate as opposed to the original ‘Films of the 40s’, ‘50s’, etc.

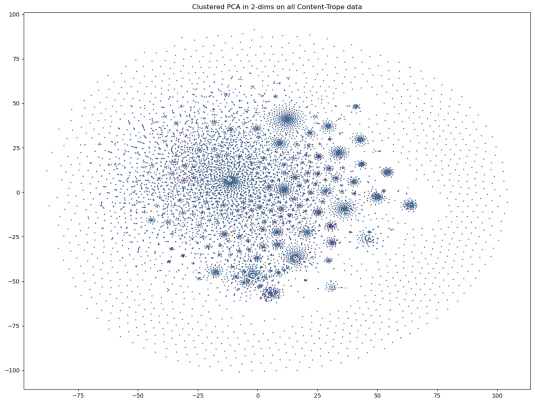




Next I examined the number of tropes present in each form of media. Most content items have less than 100 tropes present with a large number of them (~1500) having only 1. I also identified a number of super-troped media containing over 1500 documented examples of tropes.

Attempting to visualize clusters over the entire corpus of content yielded mixed results with a K-Means clustering algorithm applied to a T-SNE reduction being far more successful at distinguishing relatively equally sized clusters than the same K-Means clustering algorithm applied to PCA reduced data.





MODELING

My recommendation system, ClusterTales, contains four stages, five if you include the user interface.

A user selects a piece of media or multiple pieces of media to initialize related recommendations. Optionally they may include tropes which must mandatorily be included in the resultant recommendations. **Candidate Retrieval** occurs in which those given pieces of media are broken into their tropes and then content is surfaced which also contains those tropes. If an insufficient number of candidates is surfaced, the neighbors of those starting tropes is queried from the trope graph and the content containing the neighboring tropes is added to the list of candidates. The next stage is **Filtering** in which the original content items are removed from the list of candidates as well as any pieces of content that do not contain the user specified tropes if supplied. In **Scoring** the candidate vectors are K-Means clustered along with an amalgamated vector of the input contents. The 'home' cluster is the resultant cluster containing the amalgamation of original inputs and all other clusters are then treated as variety clusters. Lastly, **Ordering** occurs whereby the Manhattan distance from the amalgamation to each candidate is calculated, returning the 4 closest pieces of media within the home cluster and the closest representative from each variety cluster. These results are then displayed to the user in the user interface, a Streamlit app.

RESULTS

We can produce recommendations without using any user data via Tropes as an avenue for extracting domain knowledge of how the contents relate to each other. Now the quality of these recommendations is entirely subjective. The system appears to be grouping largely on the 'trope density' of content, often recommending highly troped content to other highly troped content and if the starting content is not highly troped, including the highly troped content in the variety recommendations.

CONCLUSION

ClusterTales is an interesting avenue for content recommendation and trope exploration. The recommendations may not be expected ones but this affords opportunity for serendipity to occur. One potential next step is to use Natural Language Processing to group tropes into trope topics for a more intuitive dimensionality reduction. There are also undoubtedly some performance optimizations which could increase the rapidity of recommendation.